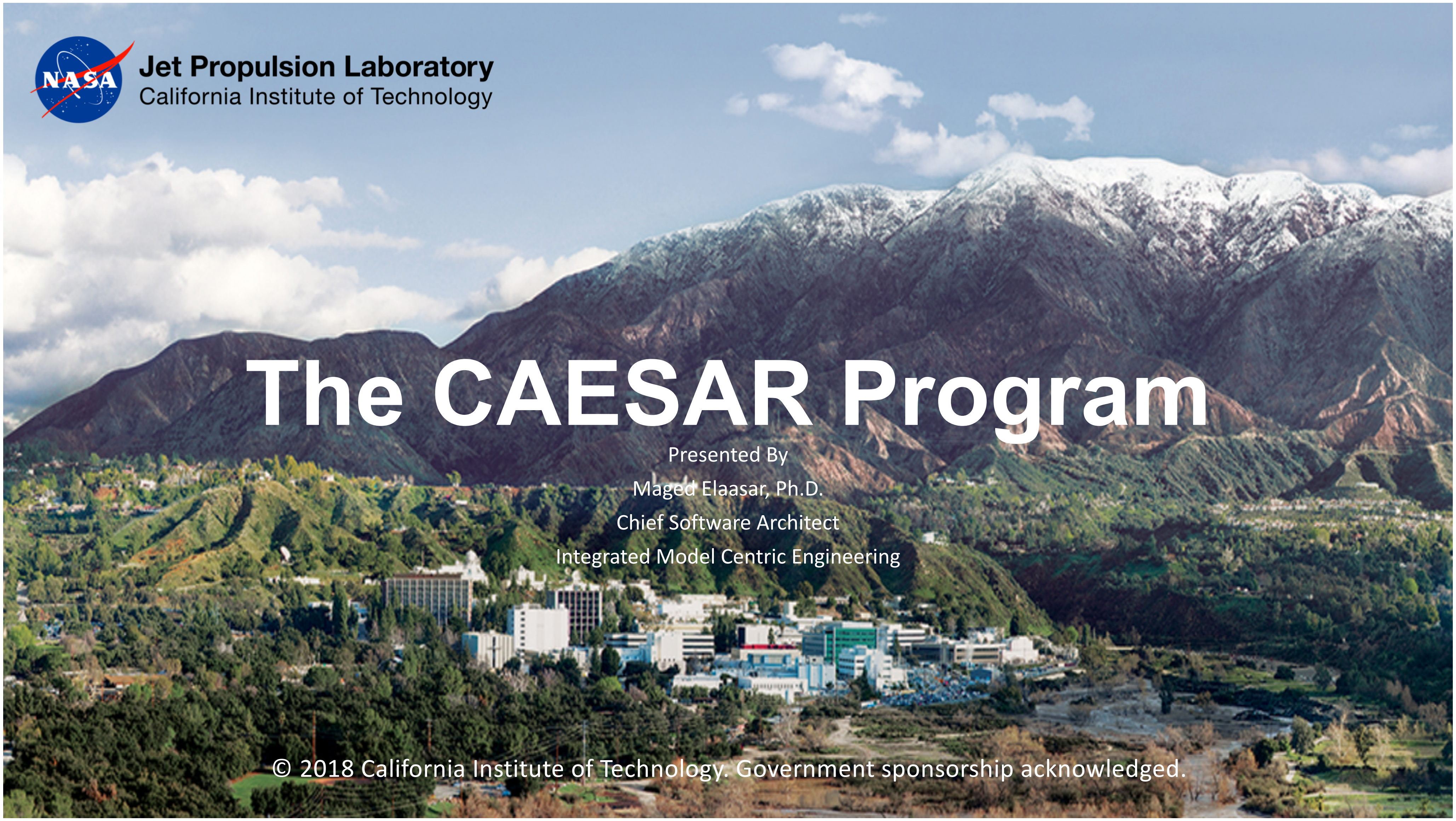


Jet Propulsion Laboratory
California Institute of Technology

The CAESAR Program

Presented By
Maged Elaasar, Ph.D.
Chief Software Architect
Integrated Model Centric Engineering

MBSE vision

- System engineering work being done in rigorous and precise way
  - Knowledge is authored, managed, and communicated through models
  - Documents are generated from models as needed for archival / presentation
- System engineering tools being federated
  - Information can be mapped, linked, and/or interchanged across tools
  - End to end use cases that span tools can be realized
- System engineering analysis being continuous and automated
  - Reduces risk and latency in decision-making
  - Improves productivity and confidence
- System engineering expertise being captured and reused
  - Shared knowledge is formalized in curated libraries and template models
  - Methodologies are rigorously defined and automated

MBSE state of practice

- Limited scope system modeling had significant success
  - Valuable baby steps demonstrated the precise and concise communication value of a model
  - Models did not attempt to cover all of the project phase details
- Ambitious system modeling effort also demonstrated value
  - But hit performance and scalability limits of trying to do everything in one tool (or one model)
- Many modeling tools are in use for system and subsystem engineering
  - Modeling tools are unconnected or weakly connected to each other
  - Performing cross-tool analysis of information is complex
  - Identifying the information baseline across tools is hard
- Focus is on tools rather than a coherent information production and management methodology
  - Most often if a methodology is used, it is ill-defined
  - Different methodologies are are being used by different projects hampering reuse
  - Some methodologies are not easily supported using the existing tools

MBSE Key challenges

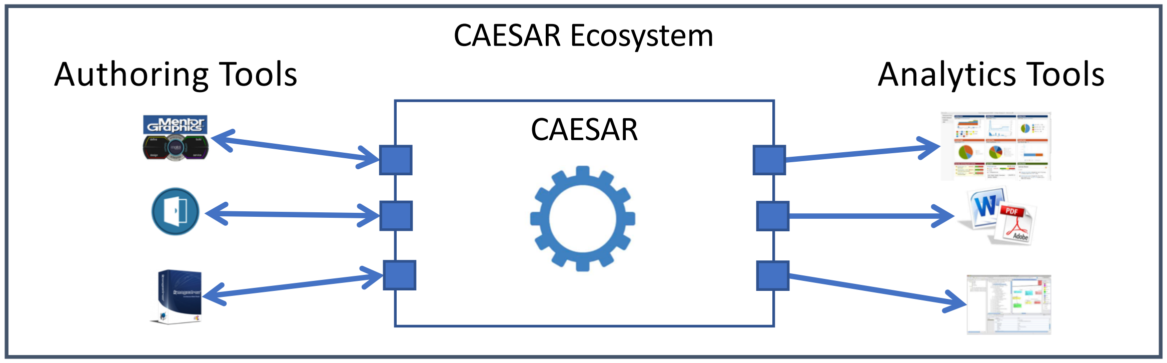
System models can be created in SysML, but they are cumbersome to create and manage	We need more domain-specific system authoring features
Heterogeneous tools and models are used in SE because many discipline-specific analyses are needed	We need effective model integration tools, methods to keep the models aligned, and to enable workflows
Model content is rapidly changing as work progresses	We need effective cross-tool configuration management, and change management
For models to be effective, it has to be easy to manage their content via meaningful documents	We need easy to use reporting tools to generate documents of model information

Solution objectives

- Develop a rigorous, scalable, extensible, multi-disciplinary, collaborative platform for systems engineering that realizes the MBSE vision and addresses its main challenges
- Develop a set of discipline-focused applications, along with their associated methodology and tool support to validate the platform and provide value to customers

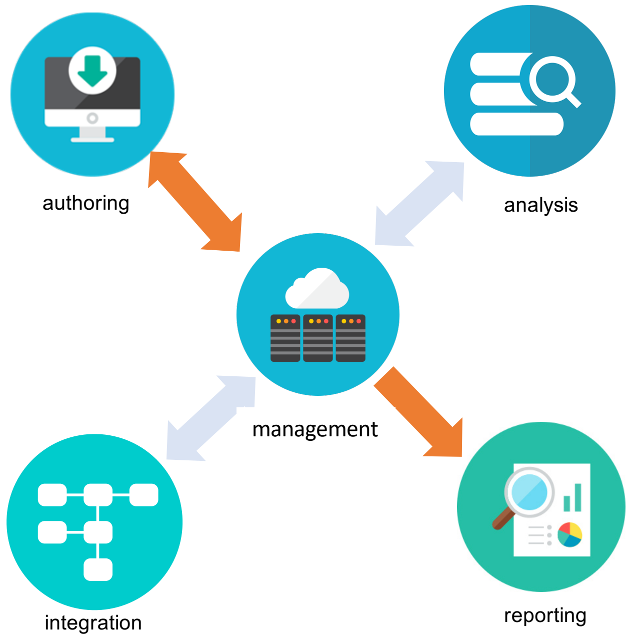
CAESAR concept

- CAESAR stands for **C**omputer **A**ided **E**ngineering for **S**ystems **AR**chitecture
- CAESAR is a platform for transforming current systems engineering practices into rigorous, integrated, model-centric engineering ones
- CAESAR is a semantic data warehouse that streamlines the practice of systems engineering



CAESAR supported functions

- Information Management
  - Common representation using rigorous semantic vocabularies
  - Configuration management, version control, and provenance
- Information Authoring
  - Support for adapting viewpoints in existing model authoring tools
  - Support for building new domain-specific model authoring viewpoints
  - Curated starting point template and library models
- Information Integration
  - Supports the design of integration flows where model fragments can be imported from different data sources in a common format and incrementally compared, analyzed, merged, and managed together
- Information Analysis
  - Support for reasoning on information
  - Support for querying model data and performing various analyses
- Information Reporting
  - Support for publishing and organizing reporting viewpoints, and generating other model representations, and documents

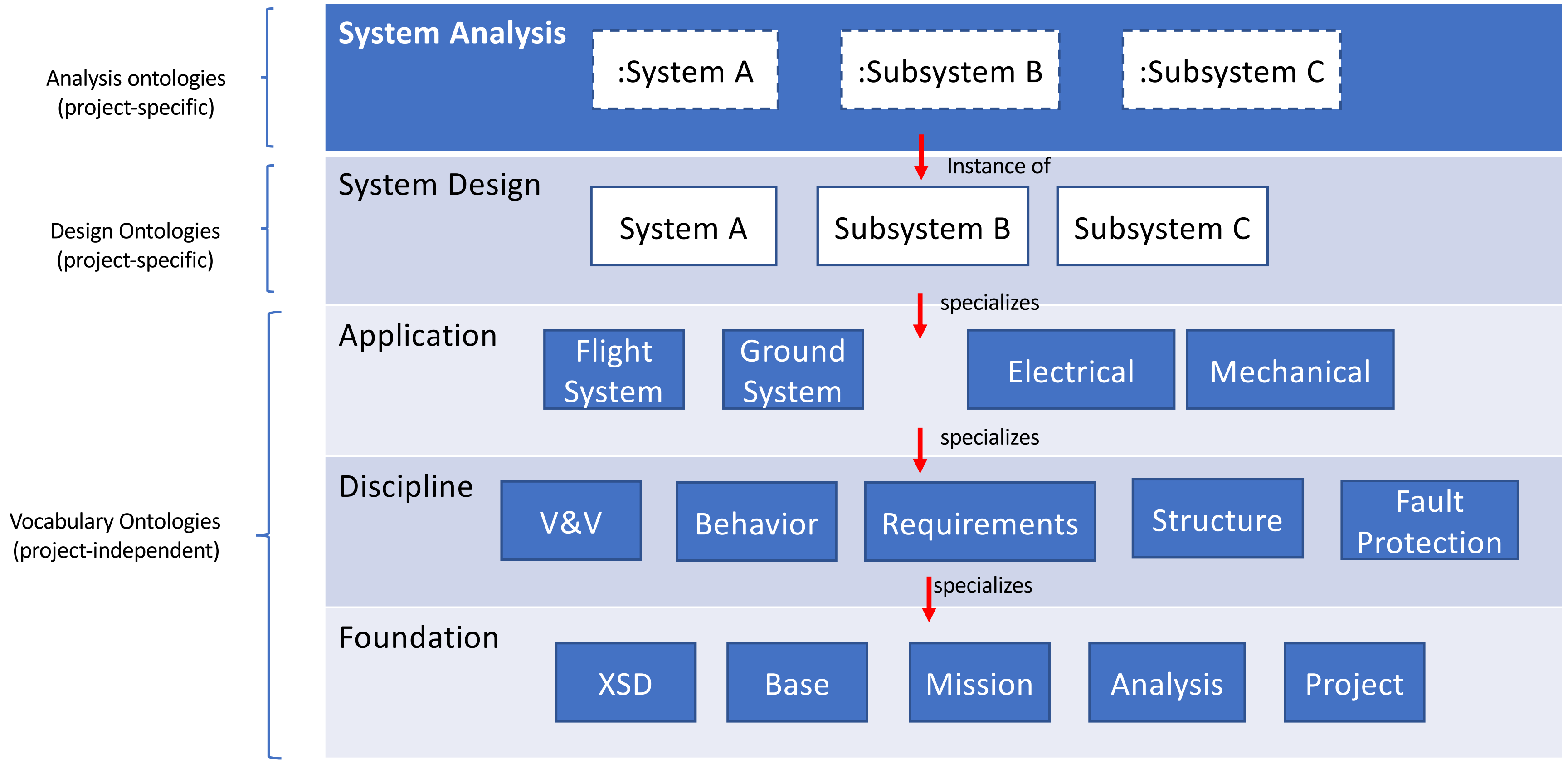


Information representation

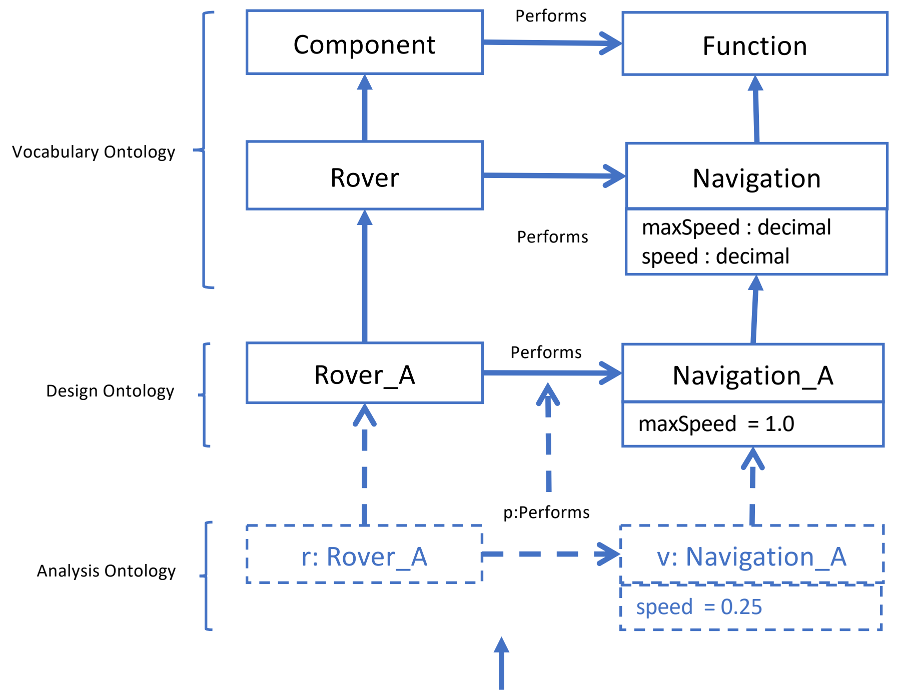
- CAESAR represents information natively in the Ontology Modeling Language (OML)
  - OML's semantics is defined by mapping to a subset of OWL2-DL plus SWRL rules
    - Allows both class-level modeling (for design) and instance-level modeling (for analysis)
    - Allows both open-world modeling (for libraries) and closed-world modeling (for systems)
    - Allows reasoning on information using over-the-shelf *Description Logic* (DL) reasoners (e.g., Pellet)
  - OML's abstract syntax is defined with EMF and encapsulates a set of patterns of using OWL2
    - Allows definition of domain-specific languages (DSLs) as layered semantic vocabularies
    - Allows information to be componentized in multiple model fragments
    - Allows the information to be expressed by different authorities (provenance)
    - Allows leveraging the large EMF ecosystem of tools
  - OML's concrete syntax consists of
    - *Textual* notation defined with Xtext (best for authoring)
    - *Graphical* notation defined with Sirius (best for visualization)
    - Compressed (tabular) notation in JSON (best for SCM)
  - OML's API consists of
    - Java API (Object-Oriented Mutable EMF API)
    - Scala API (Functional Immutable API)
    - Java Script API (Web Friendly)

OML
W3C
OWL2

OML ontology layers



OML Example



OML graphical syntax

OML textual syntax

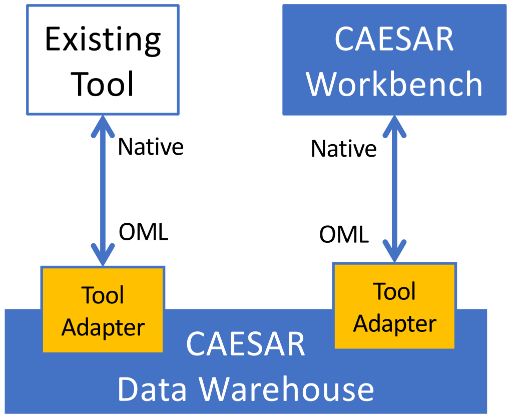
```
open terminology https://imce.jpl.nasa.gov/application/flightsystem {
   extends <https://imce.jpl.nasa.gov/foundation/mission>
   concept Rover
   Rover extendsConcept mission:Component
   concept Navigation
   Navigation extendsConcept mission:Function
   allEntities Rover . mission:Performs in Navigation
   entityScalarDataProperty maxSpeed {
      domain Navigation
      range XMLSchema:decimal
   }
   entityScalarDataProperty speed {
      domain Navigation
      range XMLSchema:decimal
   }
}

closed terminology <https://imce.jpl.nasa.gov/project/design> {
   extends <https://imce.jpl.nasa.gov/application/flightsystem>
   concept Rover_A
   Rover_A extendsConcept flightsystem:Rover
   concept Navigation_A
   Navigation_A extendsConcept flightsystem:Navigation
   allEntities Rover_A . mission:Performs in Navigation_A
   allData Navigation_A . flightsystem:maxSpeed in 1.0
}

closed description <https://imce.jpl.nasa.gov/project/analysis> {
   extends <https://imce.jpl.nasa.gov/project/design>
   conceptInstance r isA design:Rover_A
   conceptInstance v isA design:Navigation_A
   refeidRelationshipInstance p isA mission:Performs
   domain (p) = r
   range (p) = v
   v . flightsystem:speed = 0.25
}
```

Information authoring

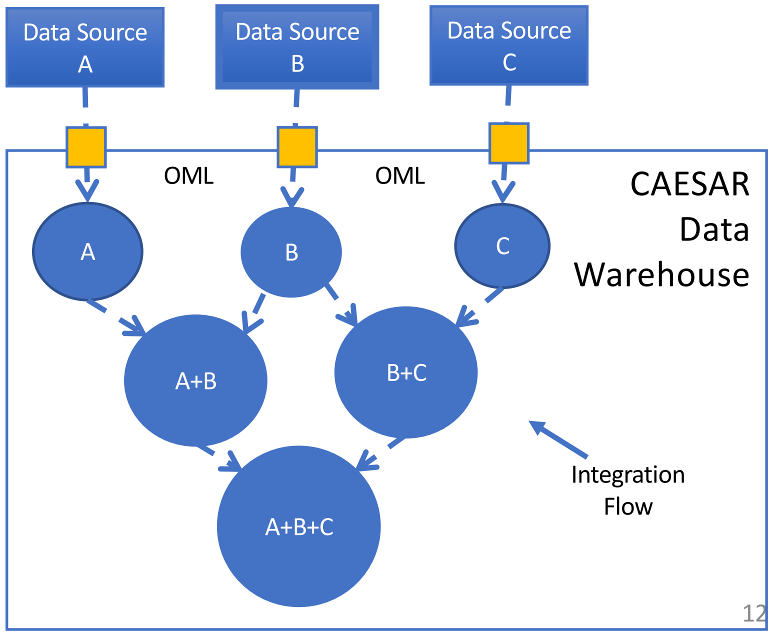
- CAESAR supports information authoring using any SE tool
  - Existing tools (COTS, open-source, in-house)
    - General purpose tools (e.g., SysML tools for system modeling)
    - Domain-specific tools (e.g., DOORS NG for requirement authoring)
  - New tools (packaged as plugins to the CAESAR Workbench)
    - Adds viewpoints that fill functional gaps in existing tools
    - Address non-functional issues (usability, scalability) of existing tools



- CAESAR provides adapters that interface with the authoring tools
  - Adapters provide data adaptation not UI adaptation
  - Adapters map between the tools native representation and OML
  - Adapters may extend the tool's native representation to increase precision
    - E.g., Represent OML vocabularies as SysML profiles in SysML tools
  - Adapters only map a subset of the information covered by the OML vocabularies
  - Adapters retrieve information for analysis and project it back for synchronization

Information integration

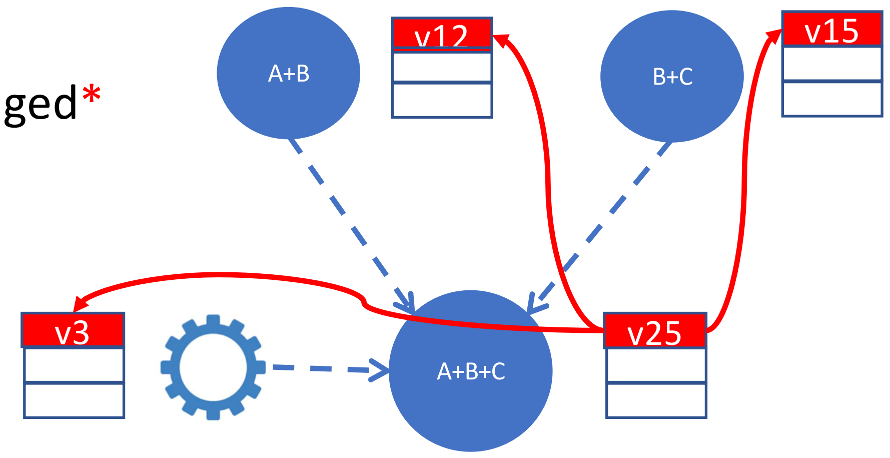
- CAESAR supports design of integration flow for information
  - Allows incremental integration of multiple data sources
    - Initial steps run the adapters to convert data into OML
      - Data is analyzed for well-formedness in silo
    - Next steps merge data from previous steps
      - data is checked for consistency with each other
  - Each integration step:
    - Configuration manages its data in an SCM repository
    - Analyzes the data using one or more analysis scripts
    - Publishes reports to allow users to inspect results
  - The integration flow
    - Can run on baseline/experimental branches of data sources
    - Can be triggered manually or automatically
    - Runs a subset of steps based on the changed data source



Information configuration management

- CAESAR configuration manages the OML data in an SCM repository
  - Data for each integration step is managed in a separate branch
  - The entire history of integration (every version) is preserved
  - Provenance information is captured as metadata with every version
    - The version of data being analyzed as input
    - The version of analysis scripts being used
  - Derived information is also configuration managed*
    - Inferred information
    - Reports and documents

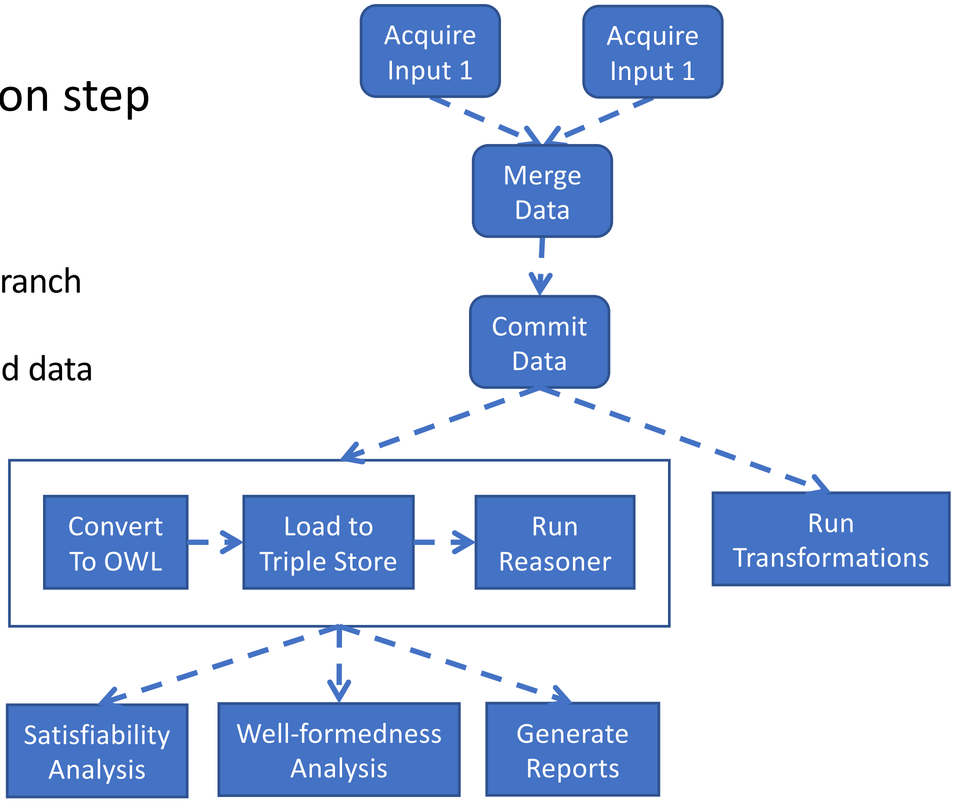
* On an SCM repo, an artifact repo, or a database



Information analysis

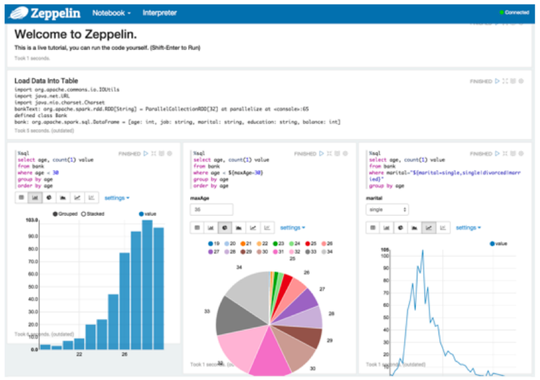
- CAESAR supports data analysis in every integration step
  - Pre-commit
    - Acquire the data to be analyzed from all inputs
  - Commit
    - A new version of the merged data to be analyzed to the branch
  - Post-commit
    - Run the configured analyses on the new version of merged data
- Different kinds of analyses can be supported:
  - Analysis using OML API
    - Run transformations
  - Analysis using OWL2-DL API*
    - Satisfiability analysis
    - Well-formedness analysis
    - Report generation

* Requires OML data to first be converted to OWL2-DL



Information reporting

- CAESAR supports generation of reports (or documents or models)
  - Supports generation of gate products during integration
    - Using report templates that query the data
      - Templates are designed using supported analytics tool
    - Can be static or interactive (allows drilling, filtering, etc.)
    - Can be organized in dashboards and made searchable
    - Can be reviewed, comment on, and approved
  - Supports generation of arbitrary products at any time
    - Using published query endpoints for every data version
    - Can be used to produce snapshot, difference, or trending reports



Welcome to Zeppelin.

This is a live tutorial, you can run the code yourself. (Shift-Enter to Run)

Load Data Into Table

```
import org.apache.commons.io.IOUtils
import java.net.URL
import java.nio.charset.Charset
bankText: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[32] at parallelize at <console>:65
defined class Bank
bank: org.apache.spark.sql.DataFrame = [age: int, job: string, marital: string, education: string, balance: int]
```

```
%sql
select age, count(1) value
from bank
where age < 30
group by age
order by age
```

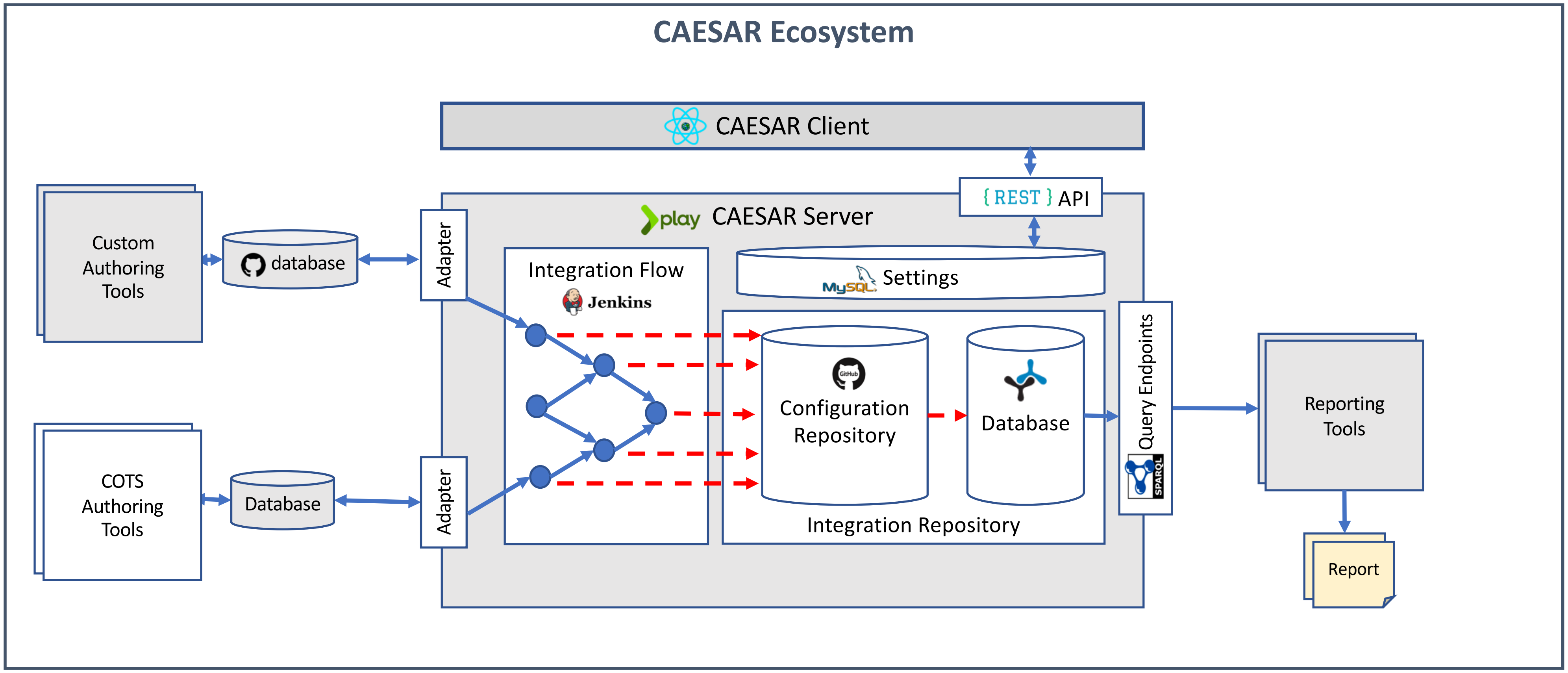
```
%sql
select age, count(1) value
from bank
where age < ${maxAge=30}
group by age
order by age
```

maxAge: 35

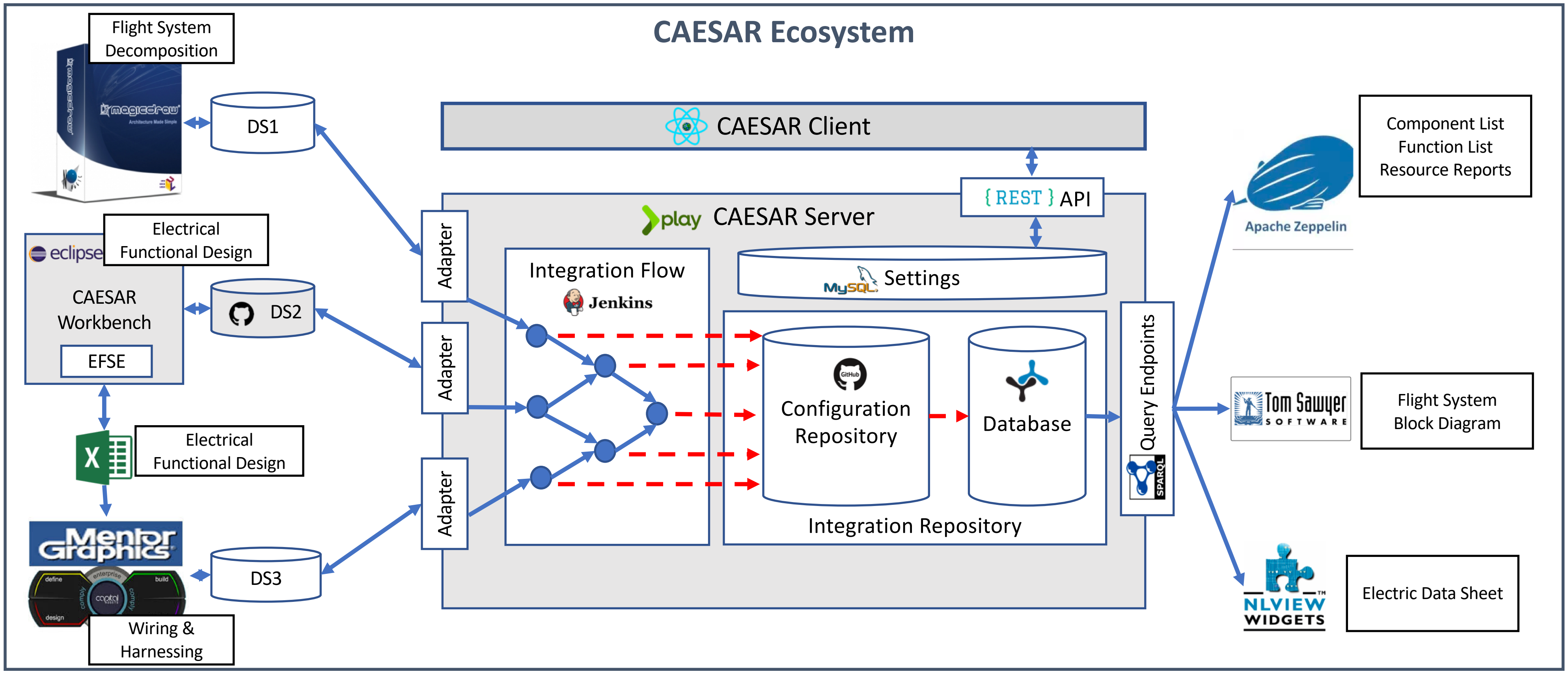
```
%sql
select age, count(1) value
from bank
where marital="${marital=single,single|divorced|married}"
group by age
```

marital: single

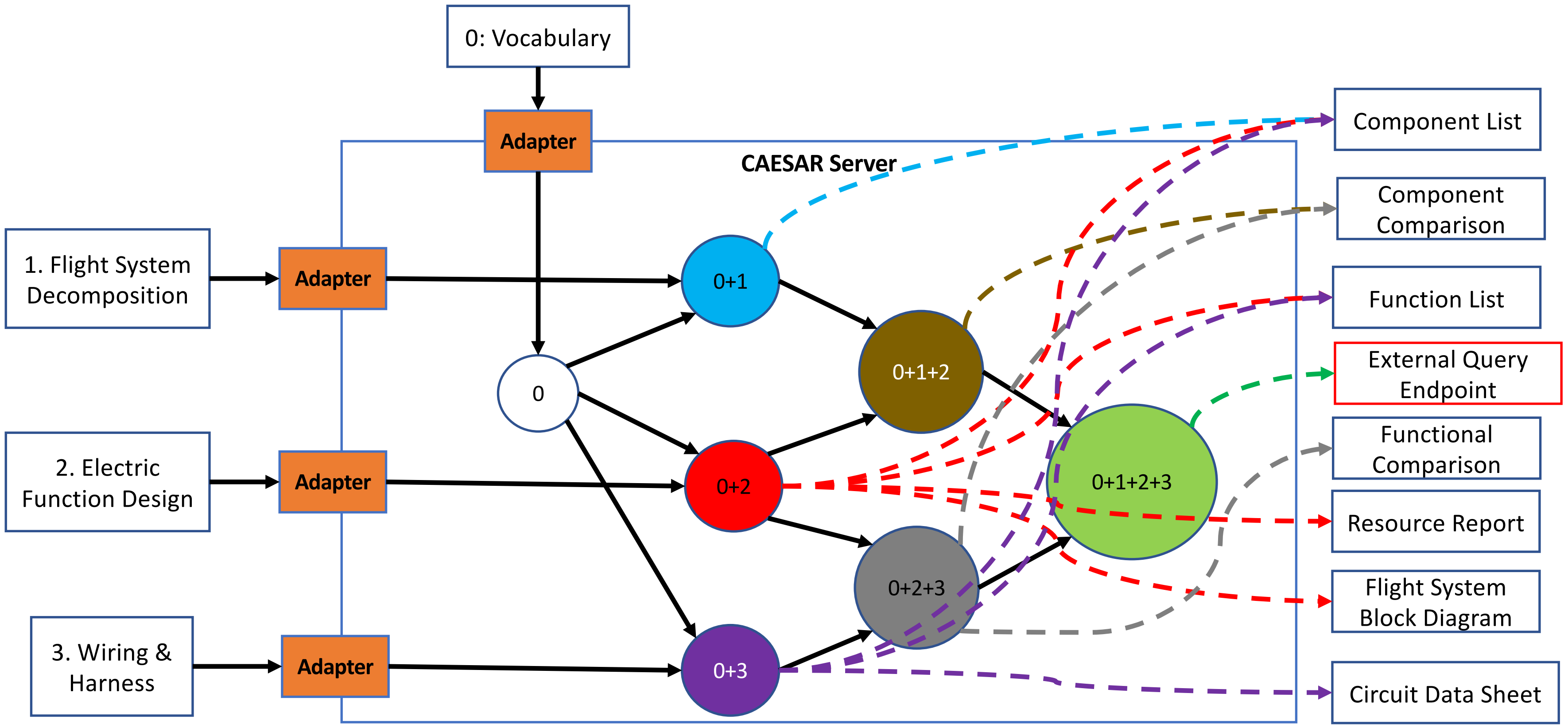
CAESAR architecture



CAESAR example application: Flight System Electrical Engineering



CAESAR example application: Flight System Electrical Engineering



CAESAR development methodology

- Work with the domain experts (the line organizations) to implement applications
  - Develop relevant ontologies
  - Identify relevant data sources (and their authoring tools)
    - Develop new authoring viewpoints if needed
  - Develop integration workflows
  - Develop analyses to run
  - Develop gate products to produce

- Work with the projects to configure the applications on the platform
  - Define data sources
  - Configure integration workflows
  - Configure query endpoints
  - Configure report templates

CAESAR development strategy

- CAESAR platform is being incrementally realized with every application
- Different CAESAR applications will be added over time
- Some components of CAESAR are already open-sourced
  - OML API and Workbench (https://github.com/JPL-IMCE/gov.nasa.jpl.imce.oml)
- Some other components will be open-sourced later
- CAESAR team is open to collaboration with partners (vendors, SMEs, users)